(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2003－36164
（Ｐ２００３－３６１６４Ａ）
(43)公開日　平成15年２月７日(2003.2.7)

(51)Int.Cl.[7]	識別記号	Ｆ Ｉ		テーマコード（参考）
G06F 3/14	350	G06F 3/14	350 Ａ	5B069
G09G 5/14		G09G 5/14	Ｃ	5C082

審査請求　未請求　請求項の数３　ＯＬ　（全７頁）

(21)出願番号　特願2001－222874(Ｐ2001－222874)

(22)出願日　平成13年７月24日(2001.7.24)

(71)出願人　000004329
日本ビクター株式会社
神奈川県横浜市神奈川区守屋町３丁目12番地

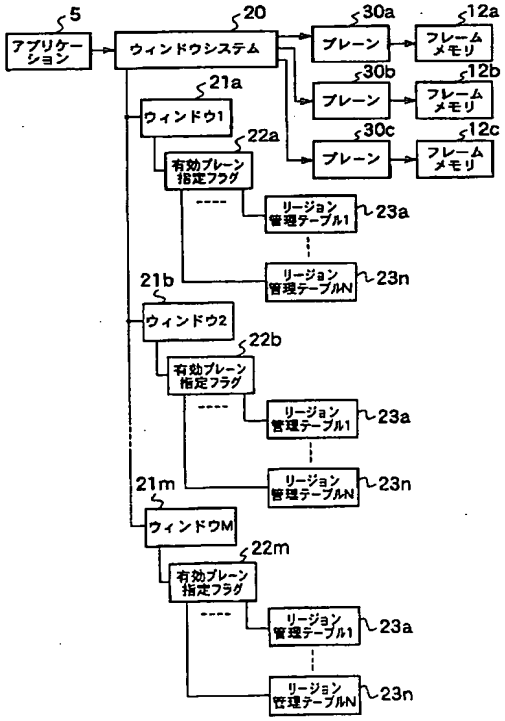
(72)発明者　大竹　信二
神奈川県横浜市神奈川区守屋町３丁目12番地　日本ビクター株式会社内

(74)代理人　100083806
弁理士　三好　秀和　（外９名）

Ｆターム(参考)　5B069 AA01 BA03 BB16 CA02 CA03
CA06 CA14 CA17
5C082 BA12 BB26 CA63 DA87 MM05

(54)【発明の名称】ウインドウシステム

(57)【要約】
【課題】　複数のグラフィク・プレーンを容易に管理することができるウインドウシステムを提供すること。
【解決手段】　アプリケーションプログラム（5）により生成される複数のウインドウ（21a-21m）を描画する複数のプレーン（30a-30c）と、前記各プレーン（30a-30c）に対応し、前記各プレーンからのデータを記憶するフレームメモリ（12a-12c）とを有し、前記各フレームメモリ（12a-12c）からの出力データを配合或いは選択して表示装置（16）に表示するウインドウシステムであって、前記生成される複数のウインドウ（21a-21m）毎に、前記複数のプレーン（30a-30c）から描画を行うプレーン（30a-30c）を選択する有効プレーン指定手段（22）を有することによる。



1

【特許請求の範囲】

【請求項１】　アプリケーションプログラムにより生成される複数のウインドウを描画する複数のプレーンと、前記各プレーンに対応し、前記各プレーンからのデータを記憶するフレームメモリとを有し、前記各フレームメモリからの出力データを配合或いは選択して表示装置に表示するウインドウシステムであって、

前記生成される複数のウインドウ毎に、前記複数のプレーンから描画を行うプレーンを選択する有効プレーン指定手段を有することを特徴とするウインドウシステム。

【請求項２】　請求項１に記載のウインドウシステムにおいて、

前記生成される複数のウインドウのうち描画が行われるプレーンを有効プレーンとして前記ウインドウが前記表示装置に表示される表示領域及び前記表示装置には表示されない非表示領域を矩形領域に分割し、各矩形領域の位置及びサイズを前記複数のウインドウの有効プレーン毎に管理するリージョン管理手段を有することを特徴とするウインドウシステム。

【請求項３】　請求項２に記載のウインドウシステムにおいて、

前記ウインドウの各矩形領域の位置或いはサイズに変化が生じた場合、再描画が必要なプレーンと再描画する矩形領域の位置及びサイズを前記アプリケーションプログラムに通知することを特徴とするウインドウシステム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は表示ウインドウを描画するための複数のプレーンを有し、各プレーンの表示データを配合あるいは選択して表示装置に表示するウインドウシステムに関する。

【０００２】

【従来の技術】グラフィク・プレーンに複数のアプリケーション画面を出力する方法として、ウインドウシステムが挙げられる。このウインドウシステムは、アプリケーションプログラムに対して、ウインドウという仮想画面を提供し、その仮想画面を１枚のグラフィク・プレーンに重ね合わせるなどして同時に複数表示することができる機能を提供するソフトウェアである。ユーザは、マウス・ポインタ等でウインドウの枠や内部をクリックすると、それがアクティブなウインドウとなり、そのアプリケーションに対して、キーボードやマウスでの入力が可能になる。

【０００３】アプリケーションプログラムは、提供されたウインドウに対して描画処理を行えば、他のアプリケーションの画面描画処理等を意識する必要はない。

【０００４】このようなウインドウシステムは多くの商品が市場に出ているが、一例を挙げると、ＵＮＩＸ（登録商標）上で動作するＸＷｉｎｄｏｗなどがある。

【０００５】これらのウインドウシステムでは、アプリケーションプログラムとの連絡のために、親ウインドウのＩＤと表示位置などの属性を引数として受け渡すウインドウ生成関数（例えば、ＸＷｉｎｄｏｗでは、関数XcreateWindow()である）を備えている。ウインドウシステムは、アプリケーションが実行するウインドウ生成関数に反応し、生成するウインドウの属性情報を格納するためのメモリを確保し、当該ウインドウの識別子をアプリケーションに返す、などの動作を行う。

【０００６】上記ウインドウ属性情報には、ウインドウの表示状態の情報も含まれ、ウインドウの表示領域のうち、実際に画面に表示されている領域と、他のウインドウが重なっているため非表示になっている領域の情報とが格納される。これらのウインドウ属性情報は、ウインドウの重なり合いやサイズなどを変更する際に、アプリケーションプログラムに対して、ウインドウの再描画を行うように再描画イベントを通知する際に用いられる。

【０００７】

【発明が解決しようとする課題】ところで、近年、複数のグラフィク・プレーンを持ち、これら複数のグラフィク・プレーンの内容をブレンド（配合）あるいは選択して表示するマルチ画面に対応した表示装置が開発されている。特に、テレビ受信機の分野ではデジタル放送が開始され、これを受けて、今後、このようなマルチ画面の機能を備えたテレビ受信機が、家庭内デジタルネットワークにおける表示装置としての役割を期待されている。

【０００８】このようなマルチ画面の機能を備えた表示装置では、映像はもちろんのこと、静止画、コンピュータグラフィックス、テキストデータなど、ＤＴＶ（DigitalTelevision）、ＤＶＤ（Digital Video Disc）、ＰＣ（Personal Computer）等から供給される様々なデジタルコンテンツを、多画面で同時に表示させることが望まれている。

【０００９】しかしながら、従来のウインドウシステムでは、複数のグラフィク・プレーンについては考慮されていない。

【００１０】従って、例えば、プレーンＡとプレーンＢの２種類のグラフィク・プレーンを備えた表示装置においても、従来のウインドウシステムではどちらか一方のグラフィク・プレーンしか管理することができない。

【００１１】ウインドウシステムを２つ起動すれば、プレーンＡとプレーンＢの両方のグラフィク・プレーンを管理することができるものの、アプリケーションは起動される２つのウインドウシステムに対する処理をそれぞれプログラミングしなければならず、開発コストの増大を招く原因となる。

【００１２】本発明はこのような事情に鑑みてなされたものであり、複数のグラフィク・プレーンを容易に管理することができるウインドウシステムを提供することを目的とする。

【００１３】

【課題を解決するための手段】上記課題を解決するために、本発明に係るウインドウシステムは、アプリケーションプログラムにより生成される複数のウインドウを描画する複数のプレーンと、前記各プレーンに対応し、前記各プレーンからのデータを記憶するフレームメモリとを有し、前記各フレームメモリからの出力データを配合或いは選択して表示装置に表示するウインドウシステムであって、前記生成される複数のウインドウ毎に、前記複数のプレーンから描画を行うプレーンを選択する有効プレーン指定手段を有することを特徴とする。

【００１４】また、前記生成される複数のウインドウのうち描画が行われるプレーンを有効プレーンとして前記ウインドウが前記表示装置に表示される表示領域及び前記表示装置には表示されない非表示領域を矩形領域に分割し、各矩形領域の位置及びサイズを前記複数のウインドウの有効プレーン毎に管理するリージョン管理手段を有することを特徴とする。

【００１５】また、前記ウインドウの各矩形領域の位置或いはサイズに変化が生じた場合、再描画が必要なプレーンと再描画する矩形領域の位置及びサイズを前記アプリケーションプログラムに通知することを特徴とする。

【００１６】本発明の特徴によれば、アプリケーションプログラムは有効プレーンを選択的に指定することができるため、アプリケーション側およびウインドウシステム側の双方は指定された有効プレーンのみを管理すれば良い。

【００１７】また、表示中の他のウインドウが移動したり閉じられたりして、今まで非表示であった領域が表示領域に変化する際にも、変化が生じたウインドウの有効プレーンがウインドウシステムから通知されるため、アプリケーションプログラムは通知されるプレーンに対してのみ再描画処理を施せば良い。

【００１８】

【発明の実施の形態】以下、図面を参照しながら、本発明の実施の形態を詳細に説明する。

【００１９】図１は本実施形態におけるウインドウシステムを動作させるためのハードウェアの構成例を模式的に示した図であり、図２はソフトウェアの構成例を模式的に示した図である。尚、図示した例では、３枚のグラフィク・プレーン３０ａ～３０ｃを備えたウインドウシステム２０を示しており、グラフィク・プレーン３０ａ～３０ｃ毎にそれぞれメモリコントローラ１３ａ，１３ｂ，１３ｃとフレームメモリ１２ａ，１２ｂ，１２ｃとを備えている。

【００２０】ＣＰＵ１０にて実行されるウインドウシステム２０は、バス１１を介して、メモリコントローラ１３ａ，１３ｂ，１３ｃにより、各グラフィク・プレーン３０ａ～３０ｃのデータをフレームメモリ１２ａ，１２ｂ，１２ｃに書き込む。

【００２１】グラフィク・プレーン３０ａのデータはメモリコントローラ１３ａによってフレームメモリ１２ａから順次読み出されて、プレーン合成コントローラ１４に転送される。同様にして、グラフィク・プレーン３０ｂ，３０ｃのデータはメモリコントローラ１３ｂ，１３ｃによってフレームメモリ１２ｂ，１２ｃから順次読み出されて、プレーン合成コントローラ１４に転送される。

【００２２】プレーン合成コントローラ１４は各フレームメモリ１２ａ，１２ｂ，１２ｃから読み出されたデータを、ブレンド（配合）あるいは選択して、Ｄ／Ａコンバータ１５に転送する。Ｄ／Ａコンバータ１５は、プレーン合成コントローラ１４から転送されてきたデータをアナログ信号に変換し、モニタ１６に送る。

【００２３】図２において、アプリケーションプログラム５は、ウインドウ生成関数を実行し、ウインドウシステム２０に対してウインドウ２１ａ～２１ｍの生成を要求する。アプリケーションプログラム５は、ウインドウ２１ａ～２１ｍの生成を要求する際に、有効プレーン指定フラグ２２ａ～２２ｍにて、グラフィク・プレーン３０ａ～３０ｃの中から該ウインドウの描画を行うプレーンを有効プレーンとして選択的に指定する。また、アプリケーションプログラム５は、生成済みのウインドウ２１ａ～２１ｍに関して、ウインドウシステム２０に対して描画、マップ等の要求も行う。

【００２４】アプリケーションプログラム５からウインドウの生成、描画、マップ等の要求を受けたウインドウシステム２０は、該ウインドウの有効プレーン毎に、該ウインドウの表示領域及び非表示領域をリージョン（REGION）と呼ばれる矩形領域に分割し、各リージョンの位置及びサイズを該ウインドウの有効プレーン毎に管理するリージョン管理テーブル２３ａ～２３ｎを作成する。そして、リージョン管理テーブル２３ａ～２３ｎの情報に基づいて、フレームメモリ１２ａ，１２ｂ，１２ｃに各グラフィク・プレーン３０ａ～３０ｃの表示データを書き込む。

【００２５】図３（ａ）は本実施形態のウインドウシステム２０におけるウインドウ生成関数CreateWindow()の指定例を示している。ウインドウ生成関数CreateWindow()は、引数として親ウインドウＩＤ（parent）と有効プレーン指定フラグ（plane）を備えており、アプリケーションプログラム５からウインドウシステム２０に対して、親ウインドウＩＤ（parent）で指定されたウインドウを親として、新規にウインドウを生成するように要求する関数である。このウインドウ生成関数CreateWindow()は、出力として、新規に生成したウインドウのＩＤをアプリケーションプログラム５に返す。

【００２６】有効プレーン指定フラグ（plane）は、図３（ｂ）に例示するように、Ｎビットのビット列から構成されている。ビット０がプレーン０、ビット１がプレーン１、ビット２がプレーン２、・・・・ビットＮがプ

レーンＮに対応しており、それぞれのビット値が「０」の場合は「無効」、「１」の場合は「有効」であることを示す。アプリケーションプログラム５は、ウインドウ生成時に複数あるグラフィク・プレーンのうち、生成するウインドウの描画を行うプレーンについて「有効」のフラグを立てる。

【００２７】図４は、ウインドウの表示されている領域と表示されていない領域とを管理するためのメモリテーブルであるリージョン管理テーブル２３の構成例を示している。このリージョン管理テーブル２３は、ウインドウの表示領域を管理する表示リージョンテーブル２４と、ウインドウの非表示領域を管理する非表示リージョンテーブル２５とからなり、生成されるウインドウの有効プレーン毎に作成される。

【００２８】表示リージョンテーブル２４は、ウインドウの表示領域を表示リージョンと呼ばれる矩形領域に分割し、分割した表示リージョン数と、分割した表示リージョン毎に始点座標（Ｘ，Ｙ）、サイズ（横幅、高さ）を記憶する。同様に、非表示リージョンテーブル２５は、ウインドウの非表示領域を非表示リージョンに分割し、分割した非表示リージョン数と、分割した非表示リージョン毎に始点座標（Ｘ，Ｙ）、サイズ（横幅、高さ）を記憶する。

【００２９】図５は、グラフィク・プレーン３０に２つのウインドウ４０，５０が表示されている例を示しており、ウインドウ５０はウインドウ４０の一部に重なっている。このウインドウ４０の表示領域は、図６に例示するように、２つの表示リージョン４１，４２に分割することができ、ウインドウ４０の非表示領域は、図７に例示するように、１つの非表示リージョン４３とすることができ、図８に示すリージョン管理テーブル２３として記憶される。

【００３０】このように、ウインドウシステム２０はウインドウの表示領域と非表示領域とをリージョン管理テーブル２３にて常に管理することで、ウインドウの描画を行う際には、有効なプレーンの表示領域にのみ描画処理を行えば良い。尚、このリージョン管理テーブル２３の各値は、ウインドウの位置やサイズ、他のウインドウとの重なり具合が変化するたびに更新される。

【００３１】図９は、２枚のグラフィク・プレーン３０，３１を備えた表示装置で、グラフィク・プレーン３０，３１を有効プレーンとしたウインドウ４０、グラフィク・プレーン３０のみを有効プレーンとしたウインドウ５０、グラフィク・プレーン３１を有効プレーンとしたウインドウ６０が表示されている様子を例示している。また、図１０はグラフィク・プレーン３０における上記各ウインドウの表示例、図１１はグラフィク・プレーン３１における上記各ウインドウの表示例を示している。

【００３２】ここで、ウインドウ５０がアンマップされた場合の、ウインドウシステム２０の動作例を説明する。アンマップされるウインドウ５０の有効プレーンは図１０に示したグラフィク・プレーン３０であるので、図１２に示すように、同じくグラフィク・プレーン３０を有効プレーンとするウインドウ４０の重なり部分（非表示領域４５）が表示領域に変化する。

【００３３】図１４は、ウインドウ５０がアンマップされた場合の動作シーケンスを例示する図である。

【００３４】まず、ウインドウ５０に対して「閉じる」操作がなされると、ウインドウ５０からウインドウシステム２０に対してアンマップ要求が通知される（Step01）。

【００３５】アンマップ要求を受けたウインドウシステム２０は、所定のアンマップに係る処理を行い、ウインドウ５０に対してアンマップを行った旨を通知する（Step02）。

【００３６】次に、ウインドウシステム２０は、ウインドウ４０に対して再描画イベントを通知する（Step03）。

【００３７】その際、再描画イベントには、図１３に示すような再描画イベント情報２６が付加される。この再描画イベント情報２６は、再描画を要するグラフィク・プレーンと、再描画を要するリージョン情報の始点座標（Ｘ，Ｙ）、サイズ（横幅、高さ）が含まれる。

【００３８】再描画イベントを受けたウインドウ４０は、再描画イベント情報２６のデータに基づいて、グラフィク・プレーン３０の指定された始点座標（Ｘ，Ｙ）とサイズ（横幅、高さ）に該当する矩形部分について再描画を行った後、ウインドウシステム２０に対して再描画完了を通知する（Step04）。

【００３９】尚、ウインドウ４０はグラフィク・プレーン３１も有効であるが、アンマップされたウインドウ５０はグラフィク・プレーン３１が有効でないので、ウインドウ４０のグラフィク・プレーン３１に関する再描画イベントは通知されない。

【００４０】また、ウインドウ６０の有効プレーンはグラフィク・プレーン３０ではないため、ウインドウ５０がアンマップされても何の影響も受けない。

【００４１】このように、アプリケーションプログラム５は有効プレーンを選択的に指定することができるため、アプリケーション５側およびウインドウシステム２０側の双方は指定された有効プレーンのみを管理すれば良い。

【００４２】また、表示中の他のウインドウが移動したり閉じられたりして、今まで非表示であった領域が表示領域に変化する際にも、変化が生じたウインドウの有効プレーンがウインドウシステム２０から通知されるため、アプリケーションプログラム５は通知されるプレーンに対してのみ再描画処理等を施せば良い。

【００４３】以上、本発明の実施形態について詳細に説

明したが、本発明は本実施例に限定されず、本発明の主旨を逸脱しない範囲において、種々の改良や変更を成し得るであろう。

【００４４】例えば、本実施形態ではＵＮＩＸのＸＷｉｎｄｏｗをモデルに説明したが、本発明はこれに限定されず、他の形式のウインドウシステムにも適用することができる。

【００４５】

【発明の効果】本発明によれば、アプリケーションプログラムは有効プレーンを選択的に指定することができるため、アプリケーション側およびウインドウシステム側の双方は指定された有効プレーンのみを管理すれば良い。

【００４６】また、表示中の他のウインドウが移動したり閉じられたりして、今まで非表示であった領域が表示領域に変化する際にも、変化が生じたウインドウの有効プレーンがウインドウシステムから通知されるため、アプリケーションプログラム５は通知されるプレーンに対してのみ再描画処理等を施せば良い。

【００４７】以上のことから、本発明によれば、複数のグラフィク・プレーンを容易に管理することができるウインドウシステムを提供することができる。

【図面の簡単な説明】

【図１】本実施形態におけるウインドウシステムのハードウェアの構成例を模式的に示した図である。

【図２】本実施形態におけるウインドウシステムのソフトウェアの構成例を模式的に示した図である。

【図３】（ａ）は本実施形態におけるウインドウシステムのウインドウ生成関数CreateWindow()、（ｂ）は有効プレーン指定フラグ（plane）の指定例を示したイメージ図である。

【図４】本実施形態におけるウインドウシステムのリージョン管理テーブルの構成例を示した図である。

【図５】本実施形態において、グラフィク・プレーンに２つのウインドウが表示されている例を示すイメージ図である。

【図６】図５に示したウインドウの表示領域を２つの表示リージョンに分割した例を示すイメージ図である。

【図７】図５に示したウインドウの非表示領域を１つの非表示リージョンとした例を示すイメージ図である。

【図８】図６～図７に示した各リージョンの情報を格納したリージョン管理テーブルの具体例である。

【図９】本実施形態において、２枚のグラフィク・プレーンを備えた表示装置で、各ウインドウが表示されている様子を例示した図である。

【図１０】図９に示したグラフィク・プレーンの表示例を示した図である。

【図１１】図９に示したグラフィク・プレーンの表示例を示した図である。

【図１２】図１０に示したグラフィク・プレーンにおいて、ウインドウの重なり部分（非表示領域）が表示領域に変化した例を示した図である。

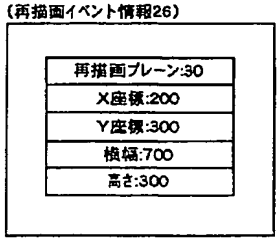
【図１３】本実施形態において、再描画イベントの通知時に付加される再描画イベント情報の具体例を示す図である。

【図１４】本実施形態における再描画イベント時の処理シーケンスを例示する図である。

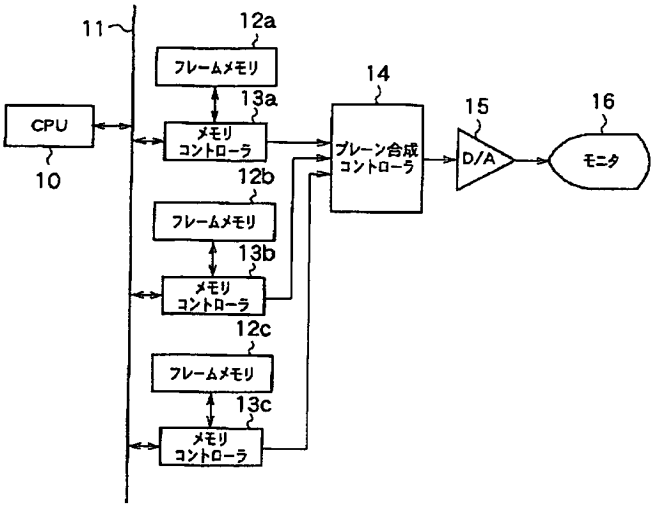
【符号の説明】

５....アプリケーションプログラム
１０....ＣＰＵ
１１....バス
１２ａ～１２ｃ....フレームメモリ
１３ａ～１３ｃ....メモリコントローラ
１４....プレーン合成コントローラ
１５....Ｄ／Ａ（デジタル／アナログ変換器）
１６....モニタ
２０....ウインドウシステム
２１ａ～２１ｍ....アドレス保持部
２２ａ～２２ｍ....有効プレーン指定フラグ
２３，２３ａ～２３ｍ....リージョン管理テーブル
２４....表示リージョンテーブル
２５....非表示リージョンテーブル
２６....再描画イベント情報
３０，３０ａ～３０ｃ，３１....グラフィック・プレーン
４０，５０，６０....ウインドウ
４１，４２....表示リージョン
４３，４５....非表示リージョン

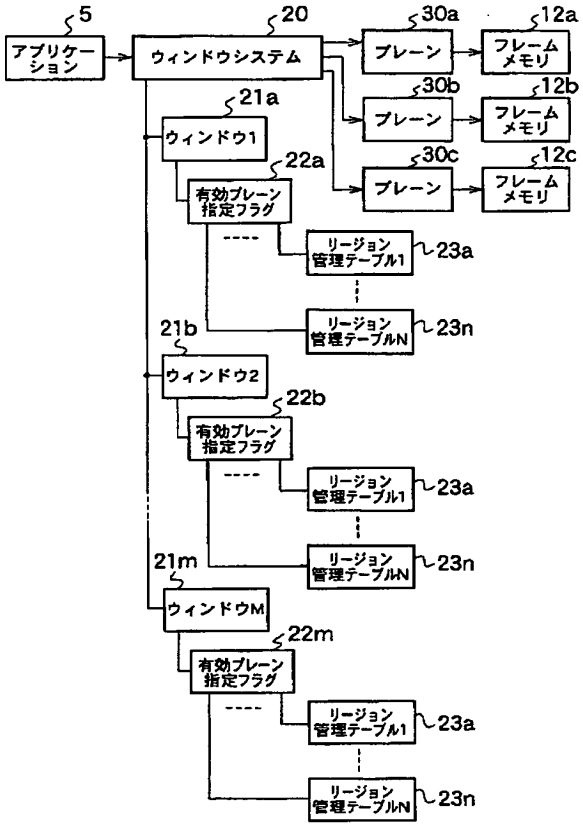
【図１３】



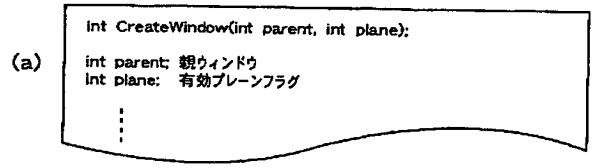
【図１】


11
12a
フレームメモリ
13a
メモリコントローラ
CPU
10
12b
フレームメモリ
13b
メモリコントローラ
12c
フレームメモリ
13c
メモリコントローラ
14
プレーン合成コントローラ
15
D/A
16
モニタ

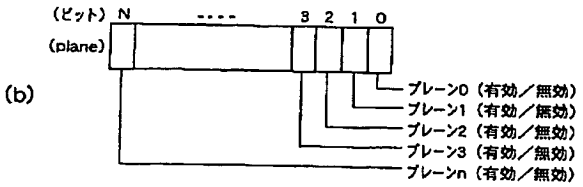
【図２】


5
アプリケーション
20
ウィンドウシステム
30a
プレーン
12a
フレームメモリ
30b
プレーン
12b
フレームメモリ
30c
プレーン
12c
フレームメモリ
21a
ウィンドウ1
22a
有効プレーン指定フラグ
リージョン管理テーブル1
23a
リージョン管理テーブルN
23n
21b
ウィンドウ2
22b
有効プレーン指定フラグ
リージョン管理テーブル1
23a
リージョン管理テーブルN
23n
21m
ウィンドウM
22m
有効プレーン指定フラグ
リージョン管理テーブル1
23a
リージョン管理テーブルN
23n

【図３】

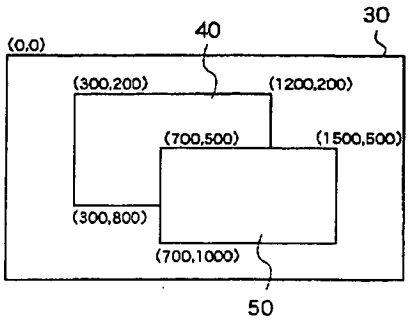

(a)
int CreateWindow(int parent, int plane);
int parent; 親ウィンドウ
int plane; 有効プレーンフラグ

【図４】

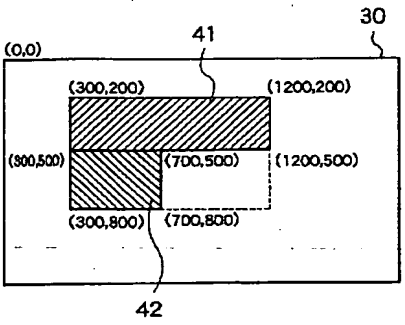

(b)
(ビット) N ---- 3 2 1 0
(plane)
プレーン0（有効／無効）
プレーン1（有効／無効）
プレーン2（有効／無効）
プレーン3（有効／無効）
プレーンn（有効／無効）


(リージョン管理テーブル23)
(表示リージョンテーブル24)
表示リージョン数 (R1+1)
表示リージョン0
X座標
Y座標
横幅
高さ
表示リージョン1
X座標
Y座標
横幅
高さ
表示リージョンR1
X座標
Y座標
横幅
高さ
(非表示リージョンテーブル25)
非表示リージョン数 (R2+1)
非表示リージョン0
X座標
Y座標
横幅
高さ
非表示リージョン1
X座標
Y座標
横幅
高さ
非表示リージョンR2
X座標
Y座標
横幅
高さ

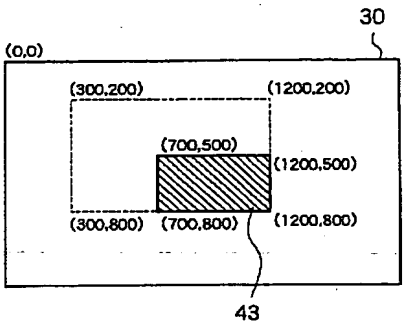
【図５】


(0,0)
40
30
(300,200)
(1200,200)
(700,500)
(1500,500)
(300,800)
(700,1000)
50

【図６】



【図７】



【図８】

(リージョン管理テーブル23)

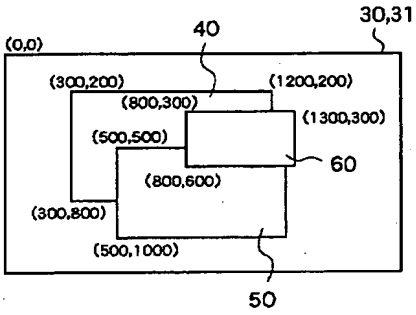
(表示リージョンテーブル24)

表示リージョン数(R1+1):2	
表示リージョン0	X座標:0
	Y座標:0
	横幅:900
	高さ:300
表示リージョン1	X座標:0
	Y座標:300
	横幅:400
	高さ:300

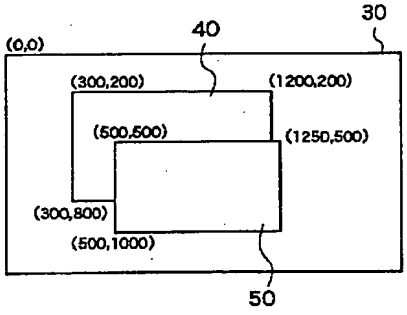
(非表示リージョンテーブル25)

非表示リージョン数(R2+1):1	
非表示リージョン0	X座標:400
	Y座標:300
	横幅:500
	高さ:300

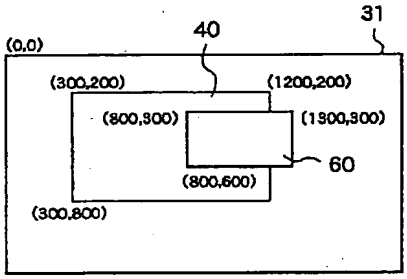
【図９】



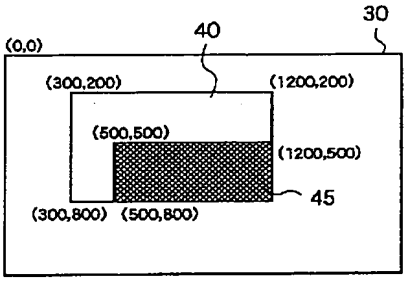
【図１０】



【図１１】



【図１２】



【図１４】